

## Temat: Podstawy JavaScript.

Serwer WWW (serwer HTTP, web server) to komputer udostępniający użytkownikom zasoby poprzez protokół http. Poprzez arkusze stylów CSS można konfigurować wygląd elementów strony internetowej po stronie klienta.

JavaScript to język skryptowy opracowany przez firmy Sun Microsystems i Netscape, przeznaczony do dynamicznej zmiany stron internetowych w przeglądarce lokalnie, po stronie klienta. Umożliwia on umieszczanie na stronach www aktywnych elementów, takich jak przyciski zmieniające wygląd, tekst przesuwany się w pasku statusu itp.

Skrypty JavaScript mogą być umieszczane wewnątrz kodu HTML (**skrypty osadzone**), w sekcjach <head> lub <body>, np.:

```
<script type="text/javascript" language="JavaScript">
  <!-- polecenia skryptu -->
</script>
```

Skrypty mogą być zapisane w odrębnych plikach (**skrypty zewnętrzne**), przyjmuje się, że mają one rozszerzenie \*.js, a wywołać je można następująco:

```
<script src="plik_javascript.js">
  <!-- polecenia skryptu -->
</script>
```

Do standardu HTML 5 dodano do znacznika script atrybuty:

```
<script src="plik_javascript.js" async> - skrypt wykonywany asynchronicznie, po pobraniu skryptu,
<script src="plik_javascript.js" defer> - skrypt zostanie wykonany dopiero po pobraniu całego kodu html.
Skrypty z atrybutami defer będą wykonywane w kolejności umieszczenia na stronie, natomiast
z atrybutami async w kolejności załadowania skryptów.
```

Znaczniki komentarza z języka HTML nie działają w obszarze skryptów. Komentarze w skryptach tworzymy w następujący sposób:

```
//komentarz w pojedynczym wierszu
/* komentarz autora skryptu w wielu wierszach */
```

Przykład kodu html strony z wywołaniem skryptu JavaScript:

```
<!DOCTYPE HTML>
<HTML lang="pl">
<HEAD>
  <TITLE>Lekcja w JavaScript</TITLE>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</HEAD>
<BODY>
  Testowanie JavaScript:<br>
  <script src="script.js"></script>
</BODY>
</HTML>
```

## Temat: Wyświetlanie komunikatów.

Przykład najprostszego pliku html z osadzonym skryptem JavaScript w sekcji <head>:

```
<!doctype html>
<html>
<head>
  <title>Skrypty JavaScript</title>
  <script language="JavaScript">
    //tekst wyświetlany w konsoli przeglądarki (Firefox - F12 lub Ctrl+Shift+I)
    console.log("To jest tekst generowany w JavaScript!")
  </script>
</head>
<body>
</body>
</html>
```

### Kolejny przykład:

```
<!DOCTYPE HTML>
<HTML lang="pl">
<HEAD>
  <TITLE> Skrypty JavaScript</TITLE>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</HEAD>
<BODY>
  Testowanie JavaScript:<br>
  <script type="text/javascript" language="JavaScript" defer>
    <!-- Komentarze języka html nie wyłączają (komentują) poleceń JavaScript
    //tekst wyświetlany w konsoli przeglądarki (F12 lub Ctrl+Shift+I)
    console.log('%c Sformatowany napis konsolowy.', 'color: red; border: 1px solid
    black; border-radius: 1em;')
    //tekst wyświetlany w przeglądarce w treści strony
    document.write('JavaScript jako treść strony.');
```

### Rodzaje okien dialogowych w języku JavaScript:

- informacyjne (`window.alert`),
- decyzyjne (`window.confirm`),
- tekstowe (`window.prompt`).

### Przykład zastosowania JavaScript do wywołania wyskakującego okienka okna informacyjnego w pliku html:

```
<!DOCTYPE HTML>
<HTML lang="pl">
<HEAD>
  <TITLE>Tomasz Greszata</TITLE>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</HEAD>
<BODY>
  Testowanie JavaScript:<br>
  <script type="text/javascript" language="JavaScript">
    window.alert(5 + 6); //window jest obiektem globalnym i można go pominąć
    alert('Witam na mojej stronie.')
```

### Inny przykład wywołania okienka informacyjnego, tym razem poprzez znacznik `<body>`:

```
<!-- wywołanie okienka powitalnego i pożegnalnego (nie wyświetla się w niektórych
przełęczarkach), polecenia JavaScript umieszczamy w deklaracji znacznika BODY -->
<BODY onload="alert('Witam na mojej witrynie!')" onunload="alert('Do następnego
spotkania!')">
```

### Przykład zastosowania JavaScript do podjęcia decyzji przez użytkownika:

```
<script type="text/javascript" language="JavaScript">
  //nie kończy procesu otwierania
  confirm('Czy chcesz wyświetlić stronę?')
```

### Przykład zastosowania JavaScript do wprowadzenia danych przez użytkownika:

```
<script type="text/javascript" language="JavaScript">
  //okno pobierające wartość od użytkownika
  let imie = prompt('Podaj swoje imię: ');
  document.write('Witaj '+imie);
```

### Przykład zastosowania w pliku html w osadzonym skrypcie JavaScript znaczników html:

```
<script type="text/javascript" language="JavaScript">
```

```
document.write('<h1>Napis jako treść strony z <b> JavaScript</b>.</h1><br>');
document.write('<i>Do formatowania napisów można stosować znaczniki
<b>html</b>.</i><br>');
</script>
```

**Konsolę w przeglądarce Firefox** można wywołać wybierając opcję Narzędzia => Konsola błędów lub klawiszem F12 lub kombinacją klawiszy Ctrl+Shift+I. Przykładowe polecenia wydane w konsoli:

```
//<script>
//podstawowe funkcje, przypisywanie wartości zmiennym, Ctrl+Shift+I
console.log("Witaj."); //napis wyświetlany w konsoli przeglądarki
console.log("Witaj.") //średnik nie jest wymagany, ale zalecany
const liczba = 31;
console.log(liczba);
liczba = 13; //przeglądarka zgłosi błąd, nie można zmienić stałej
var x; //deklaracja zmiennej
console.log(x);
x = 12; //operator przypisania
console.log(x);
x = 3.13;
console.log(x);
x = 'Witam\`ci\`nę.'; //pozwala użyć znaków specjalnych (` ,tab, enter)
console.log(x);
x = true;
console.log(x);
x = false;
console.log(x);
x = null;
console.log(x);
//</script>
```

## Temat: Zmienne i ich typy.

Typy danych w języku JavaScript:

- typ liczbowy – Number (**3.14**),
- typ łańcuchowy – ciąg znaków zamknięty znakami cudzysłowów lub apostrofów,
- typ logiczny – TRUE i FALSE,
- typ obiektowy - OBJECT,
- typ tablicowy - ARRAY,
- typ NULL,
- typ UNDEFINED.

Zadanie:

Zapoznaj się z opisem zmiennych stosowanych w języku JavaScript publikowanym na witrynie internetowej serwisu w3schools: [https://www.w3schools.com/js/js\\_variables.asp](https://www.w3schools.com/js/js_variables.asp)

Zmienne przechowują dane i wyniki. Nazwy zmiennych nie mogą zaczynać się od cyfr, nie mogą zawierać spacji oraz nie mogą wykorzystywać słów kluczowych. W nazwach stosujemy litery małe i duże, cyfry, znak \$ oraz podkreślenie `_`. Zmienne mogą zmieniać swoje wartości w trakcie wykonywania skryptu. Zmienne deklarujemy przy użyciu słowa kluczowego `var`.

```
<script>
var tekst="Twoja liczba wynosi"; //definicja zmiennej globalnej
document.writeln(typeof tekst); //wyświetlenie informacji o typie zm.
let liczba=14; //definicja zmiennej blokowej
{ //wyodrębniony blok instrukcji
  let liczba=7;
  document.writeln(tekst+" "+liczba+"<br>");
}
document.writeln(tekst+" "+liczba+"<br>");
var x=3; //definicja zmiennej liczbowej
var y=prompt("Podaj wartość y:"); //wartość traktowana jako tekst
var z=Number(prompt("Podaj wartość z:")); //wartość traktowana jako liczba
var suma; //deklaracja zmiennej
```

```

suma=x+y;
document.write('<br>Suma x, y, wynosi '+suma);
suma=x+z;
document.write('<br>Suma x, z, wynosi '+suma);
suma=x+Number(y); //konwertowanie do liczby całkowitej
document.write('<br>Suma x, y, wynosi '+suma);
suma=x+parseInt(y); //parsowanie do liczby całkowitej
document.write('<br>Suma x, y, wynosi '+suma);
suma=x+parseFloat(y); //parsowanie do liczby zmiennoprzecinkowej
document.write('<br>Suma x, y, wynosi '+suma);
</script>

```

W skryptach JavaScript można używać stałych, których wartości nie mogą ulec zmianie.

```

<script>
const pi = 3.1415; //definicja stałej
document.write('Wartość PI wynosi '+pi);
</script>

```

## Temat: Operatory w JavaScript.

### Zadanie:

Zapoznaj się z opisem operatorów stosowanych w języku JavaScript publikowanym na witrynie internetowej serwisu w3schools: [https://www.w3schools.com/js/js\\_operators.asp](https://www.w3schools.com/js/js_operators.asp)

### Operatory arytmetyczne: +, -, \*, /, \*\* (potęgowanie) oraz % (dzielenie modulo).

```

<script>
var x=13;
var y=5;
document.write(x+' + '+y+' = '+ (x+y)+"<br>");
document.write(x+' - '+y+' = '+ (x-y)+"<br>");
document.write(x+' * '+y+' = '+ (x*y)+"<br>");
document.write(x+' / '+y+' = '+ (x/y)+"<br>");
document.write(x+' ** 2 = '+ (x**2)+"<br>");
document.write(x+' % '+y+' = '+ (x%y)+"<br>");
</script>

```

### Ćwiczenie:

Napisz kod w języku JavaScript, który obliczy średnią arytmetyczną dwóch liczb podanych przez użytkownika.

### Operatory porównania (wynikiem jest true lub false):

!=	2!=3	true
==	2=="2"	true
===	2===2	false //wartości nie są tego samego typu
>	2>3	false
>=	2>=3	false
<	2<3	true
<=	2<=3	true

### Przykłady zastosowania operatorów porównania:

```

<script>
var x=9;
var y=5;
document.write('Wynikiem porównania x &lt; y jest: '+ (x<y)+"<br>");
var a=2;
var b="2";
document.write('Wynikiem porównania a == b jest: '+ (a==b)+"<br>");
document.write('Wynikiem porównania a === b jest: '+ (a===b)+"<br>");
</script>

```

### Operatory logiczne:

&&	koniunkcja (AND)	false&&true => false (średni priorytet)
	alternatywa (OR)	false  true => true (najniższy priorytet)
!	negacja (NOT)	!true => false (najwyższy priorytet)

**Przykłady zastosowania operatorów logicznych:**

```

<script>
  //typ logiczny (boolean)
  let firstValue = true
  let secondValue = false
  //koniunkcja (wynik false)
  let addResult = firstValue && secondValue
  console.log(addResult)
  //alternatywa (wynik true)
  let orResult = firstValue || secondValue
  console.log(orResult)
  //negacja (wynik false)
  let notResult = !firstValue
  console.log(notResult)
  let longExpression = firstValue || secondValue && notResult || !orResult
  console.log(longExpression)
  let longExpression = (firstValue || secondValue) && (notResult || !orResult)
  console.log(longExpression)
</script>

```

**Operatory bitowe:**

```

&    AND    1&1 => 1
^    XOR    1^1 => 0
|    OR     1|1 => 1
<<   przesunięcie w lewo o podaną liczbę miejsc
>>   przesunięcie w prawo o podaną liczbę miejsc

```

**Przykład zastosowania:**

```

<script>
  var x=9;
  var y=13;
  document.write('Liczba '+x+" binarnie to = 1001<br>");
  document.write('Liczba '+y+" binarnie to = 1101<br>");
  document.write('9 & 13 '+ (x&y)+" czyli 1001<br>");
  document.write('9 ^ 13 '+ (x^y)+" czyli 0100<br>");
  document.write('9 | 13 '+ (x|y)+" czyli 1101<br>");
  document.write('9 << 2 '+ (x<<2)+" czyli 100100<br>");
</script>

```

**Operatory przypisania:**

```

+=      a+=b      a=a+b
-=      a-=b      a=a-b
*=      a*=b      a=a*b
/=      a/=b      a=a/b
%=      a%=b      a=a%b      //dzielenie modulo (reszta z dzielenia)
++      a++       a=a+1      //postinkrementacja
++      ++a       a=a+1      //preinkrementacja
--      a--       a=a-1      //postdekrementacja
--      --a       a=a-1      //predekrementacja

```

**Przykład zastosowania operatorów przypisania:**

```

<script>
  var a=9;
  var b=13;
  var c=11;
  document.write('Wartości początkowe a='+a+" b="+b+" c="+c+"<br>");
  a+=b;
  c*=b;
  document.write('Wartości po przypisaniu a+=b =' +a+" c*=b =" +c+"<br>");
  document.write('Wartość początkowa a =' +a+"<br>");
  document.write('Wartość postinkrementacji a =' +(a++)+"<br>");
  document.write('Wartość w kolejnej linii kodu a =' +a+"<br>");
  document.write('Wartość preinkrementacji a =' +(++a)+"<br>");
  document.write('Wartość w kolejnej linii kodu a =' +a+"<br>");

```

```
</script>
```

Przykładowe operacje matematyczne dostępne w obiekcie Math:

```
<script>
  console.log(Math.floor(3,6)); //zaokrąglenie w dół => 3
  console.log(Math.round(3,6)); //zaokrąglenie => 4
  console.log(Math.abs(-3,6)); //wartość bezwzględna => 3,6
</script>
```

## Temat: Instrukcje warunkowe w JavaScript.

Instrukcje warunkowe wykonują określone polecenia w zależności od spełnienia warunków logicznych. W języku JavaScript wyróżniamy instrukcje warunkowe `if` oraz `switch`.

Zadanie:

Zapoznaj się z opisem instrukcji warunkowej `if` w języku JavaScript publikowanym na witrynie internetowej serwisu `w3schools`: [https://www.w3schools.com/js/js\\_if\\_else.asp](https://www.w3schools.com/js/js_if_else.asp)

Ogólna postać niepełnej instrukcji warunkowej `if`:

```
if(warunek)
{
  instrukcje;
}
```

Przykład zastosowania niepełnej instrukcji warunkowej `if`:

```
<script type="text/javascript" language="JavaScript">
  var x = 1; //zmienna globalna
  let y = 1; //zmienna blokowa
  if (true)
  {
    var x = 2;
    let y = 2;
  }
  console.log(x);
  console.log(y);
</script>
```

Inny przykład zastosowania niepełnej instrukcji warunkowej `if`:

```
<script>
  var liczbaNumber(prompt("Podaj liczbę: "));
  if(liczba%2==0)
  {
    alert("Podana liczba jest parzysta: "+liczba);
  }
</script>
```

Ogólna postać pełnej instrukcji warunkowej `if`:

```
if(warunek)
{
  instrukcje1;
}
else
{
  instrukcje2;
}
```

Przykład zastosowania pełnej instrukcji warunkowej `if`:

```
<script>
  var liczba=Number(prompt("Podaj liczbę: "));
  if(liczba%2==0)
  {
    alert("Podana liczba jest parzysta: "+liczba);
  }
}
```

```
    else
    {
        alert("Podana liczba jest nieparzysta: "+liczba);
    }
</script>
```

Przykład zastosowania instrukcji przetwarzania warunkowego ? do rozwiązania powyższego zadania:

```
<script>
    var liczba=Number(prompt("Podaj liczbę: "));
    var test=(liczba%2==0)? "parzysta":"nieparzysta";
    alert("Podana liczba jest "+test);
</script>
```

Inny przykład zastosowania instrukcji przetwarzania warunkowego ?:

```
<script>
    let wiek=Number(prompt("Podaj ile masz lat: "));
    wiek >=18 ? alert("Jesteś pełnoletni."):alert("Jesteś niepełnoletni.");
</script>
```

Inny przykład zastosowania pełnej instrukcji warunkowej if:

```
<script type="text/javascript">
    var imie = prompt ("Podaj swoje imię:", "");
    if (imie == null)
    {
        document.write("<h3>Dlaczego nie chcesz podać swojego imienia?</h3>");
    }
    else
    {
        document.write("<h3>Witaj " + imie + "!</h3>");
    }
</script>
```

Inny przykład zastosowania pełnej instrukcji warunkowej if:

```
<script>
    if(confirm("Podejmij decyzję!"))
    {
        alert("Kliknięto: OK");
    }
    else
    {
        alert("Kliknięto: Anuluj");
    }
</script>
```

Ogólna postać instrukcji warunkowej wielokrotnego sprawdzania else if:

```
if(warunek1)
{
    instrukcje1;
}
else if (warunek2)
{
    instrukcje2;
}
else if (warunek3)
{
    instrukcje3;
}
else
{
    instrukcje4;
}
```

Przykład zastosowania instrukcji warunkowej wielokrotnego sprawdzania else if:

```
<script type="text/javascript">
    document.write("Skrypt sprawdza znak podanej przez użytkownika liczby.");
</script>
```

```
var a = prompt("Podaj liczbę: ");
//konwertowanie zmiennej na liczbę
a = Number(a);
if (a > 0) {
    document.write("<br>Liczba " + a + " jest dodatnia.");
}
else if (a < 0)
{
    document.write("<br>Liczba " + a + " jest ujemna.");
}
else
{
    document.write("<br>Liczba " + a + " jest równa 0.");
}
</script>
```

Wykonaj zadanie:

Zmodyfikuj powyższy skrypt do poprawnego działania w przypadku, gdy użytkownik poda niewłaściwe dane, np. tekst zamiast liczby.

Wykonaj zadanie1:

Napisz skrypt osadzony w pliku html w języku JavaScript sprawdzający, która z dwóch liczb podanych przez użytkownika jest większa. W skrypcie zastosuj treści informujące o temacie zadania oraz komunikat wynikowy wyświetlony na stronie tekstem sformatowanym jako nagłówek H3 w kolorze niebieskim. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_wieksza.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

Inny przykład zastosowania instrukcji warunkowej wielokrotnego sprawdzania else if:

```
<script type="text/javascript">
    document.write("Skrypt sprawdza największą liczbę z trzech podanych przez
użytkownika.<br>");
    var a = Number(prompt("Podaj pierwszą liczbę: "));
    var b = Number(prompt("Podaj drugą liczbę: "));
    var c = Number(prompt("Podaj trzecią liczbę: "));
    document.write("Z pośród podanych liczb: " +a+", "+b+", " oraz "+c+",<br>");
    if (a >= b && a>=c) {
        document.write("największą liczbą jest: " + a);
    }
    else if (b >= a && b>=c) {
        document.write("największą liczbą jest: " + b);
    }
    else {
        document.write("największą liczbą jest: " + c);
    }
</script>
```

Wykonaj zadanie2:

Napisz skrypt osadzony w pliku html w języku JavaScript sprawdzający, czy z podanych przez użytkownika trzech długości odcinków można zbudować trójkąt. Komunikat wynikowy ma być wyświetlony na stronie tekstem sformatowanym jako nagłówek H3 w kolorze czerwonym dla wyniku negatywnego, a w kolorze zielonym dla wyniku pozytywnego. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_trojkat.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

Zadanie:

Zapoznaj się z opisem instrukcji switch w języku JavaScript publikowanym na witrynie internetowej serwisu w3schools: [https://www.w3schools.com/js/js\\_switch.asp](https://www.w3schools.com/js/js_switch.asp)

Instrukcja wyboru switch pozwala sprawdzić wiele warunków w wygodniejszy sposób. Instrukcja porównuje wyrażenie do wymienionych wartości i w przypadku uzyskania prawdy, wykonuje wymienione instrukcje



i przerywa działanie instrukcji `switch`. W przypadku braku spełnienia któregoś z warunków, wykonane zostaną instrukcje z bloku `default`.

Ogólna postać instrukcji `switch`:

```
switch(wyrażenie)
{
  case wartość1:
    instrukcje1;
    break;
  case wartość2:
    instrukcje2;
    break;
  default:           //opcjonalnie
    instrukcje3;
}
```

Przykład zastosowania instrukcji warunkowej wielokrotnego sprawdzania `switch`:

```
<script>
  document.write("Skrypt wykonuje wybrane działanie arytmetyczne na dwóch liczbach
  podanych przez użytkownika.<br>");
  var a = prompt("Podaj pierwszą liczbę: ");
  var b = prompt("Podaj drugą liczbę: ");
  //konwertowanie zmiennych na liczby
  a = Number(a);
  b = Number(b);
  var dzialanie = prompt("Podaj dzialanie (+, -, *, /): ");
  switch(dzialanie)
  {
    case '+':
      document.write("Wybrałeś dodawanie: "+a+" + "+b+" = "+(a+b));
      break;
    case '-':
      document.write("Wybrałeś odejmowanie: "+a+" - "+b+" = "+(a-b));
      break;
    case '*':
      document.write("Wybrałeś mnożenie: "+a+" * "+b+" = "+(a*b));
      break;
    case '/':
      document.write("Wybrałeś dzielenie: "+a+" / "+b+" = "+(a/b));
      break;
    default:
      document.write("Dokonałeś niewłaściwego wyboru działania: "+dzialanie);
      break;
  }
</script>
```

Inny przykład zastosowania instrukcji warunkowej wielokrotnego sprawdzania `switch`:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
  <title>Skrypty JavaScript</title>
</head>
<body>
<p id="skrypt" align="center"></p>
<script>
  let dzien;
  switch (new Date().getDay())
  {
    case 0:
      dzien = "niedziela";
      break;
    case 1:
      dzien = "poniedziałek";
      break;
    case 2:
      dzien = "wtorek";
      break;
  }
```

```
    case 3:
        dzien = "środa";
        break;
    case 4:
        dzien = "czwartek";
        break;
    case 5:
        dzien = "piątek";
        break;
    case 6:
        dzien = "sobota";
    }
    document.getElementById("skrypt").innerHTML="<h3>Skrypt  sprawdza  bieżący  dzień
tygodnia.<br>Dzisiaj jest: "+dzien+"</h3>";
</script>
</body>
</html>
```

Wykonaj zadanie:

Napisz skrypt osadzony w pliku html w języku JavaScript z wykorzystaniem instrukcji `if` realizujący to samo, co w powyższym przykładzie. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_dzien.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
    <title>Skrypty JavaScript</title>
</head>
<body>
<p id="skrypt" align="center">
<script>
    document.getElementById("skrypt").innerHTML="<h3>Skrypt  sprawdza  bieżący  dzień
tygodnia.</h3>";
    let dzien=(new Date().getDay());
    if (dzien==0) {
        document.write("Dzisiaj jest niedziela.");
    }
    else if (dzien==1) {
        document.write("Dzisiaj jest poniedziałek.");
    }
    else if (dzien==2) {
        document.write("Dzisiaj jest wtorek.");
    }
    else if (dzien==3) {
        document.write("Dzisiaj jest środa.");
    }
    else if (dzien==4) {
        document.write("Dzisiaj jest czwartek.");
    }
    else if (dzien==5) {
        document.write("Dzisiaj jest piątek.");
    }
    else if (dzien==6) {
        document.write("Dzisiaj jest sobota.");
    }
</script>
</p>
</body>
</html>
```

**Temat: Pętla for w JavaScript.**

Zadanie:

Zapoznaj się z opisem pętli for w języku JavaScript publikowanym na witrynie internetowej serwisu w3schools: [https://www.w3schools.com/js/js\\_loop\\_for.asp](https://www.w3schools.com/js/js_loop_for.asp)

**Pętla for** powtarza zestaw instrukcji określoną przez licznik ilość razy. Ogólna postać instrukcji for:

```
for (start; koniec; krok;)
{
    instrukcje1;
}
```

Przykład zastosowania pętli for:

```
<p align="center">
<script>
for (var i=1; i<=5; i++)
{
    //ciało pętli
    document.write("Stan licznika pętli: " + i + "<br>");
}
</script>
</p>
```

Inny przykład zastosowania pętli for:

```
<p id="skrypt" align="center"></p>
<script>
for (var i=5; i>=1; i--)
{
    //ciało pętli
    document.getElementById("skrypt").innerHTML += "Licznik " + i + "<br>";
}
</script>
```

Inny przykład zastosowania pętli for:

```
<p id="skrypt" align="center"></p>
<script>
<div id="instrukcja">Skrypt obliczający sumę pierwszych 10 liczb
naturalnych.<br><br></div>
var suma = 0;
for (var i=1; i<=10; i++)
{
    suma+=i;
}
document.getElementById("skrypt").innerHTML += "Suma liczb wynosi "+suma;
</script>
```

Inny przykład zastosowania pętli for:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
<title>Skrypty JavaScript</title>
<meta charset="utf-8">
<style>
body
{
background-color: #000000;
color: lightgray;
font-size: 20px;
font-family: 'Calibri';
font-weight: 700;
}
</style>
</head>
<body>
<div id="instrukcja" align="center">Skrypt wyświetlający hasło i ukrywający
je.<br><br></div>
<div id="napis" align="center"></div>
<div id="wynik" align="center"></div>
<script type="text/javascript">
var haslo = "Nauka to potęgi klucz";
```

```
//haslo = haslo.toUpperCase();
var dlugosc = haslo.length;
var hasloukryte = "";
for (i=0; i<dlugosc; i++)
{
    //if (haslo[i]==" ") hasloukryte = hasloukryte + " ";
    if (haslo.charAt(i)==" ") hasloukryte = hasloukryte + " ";
    else hasloukryte = hasloukryte + "-";
}
document.getElementById("napis").innerHTML = haslo;
document.getElementById("wynik").innerHTML = hasloukryte;
</script>
</body>
</html>
```

### Wykonaj zadanie:

Zmodyfikuj skrypt w powyższym przykładzie tak, aby hasło było wprowadzane przez użytkownika. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_haslo.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

### Inny przykład zastosowania pętli for:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
    <title>Skrypty JavaScript</title>
    <meta charset="utf-8">
</head>
<body>
    <div id="instrukcja" align="center">Skrypt wyświetlający liczby parzyste z przedziału liczb podanych przez użytkownika.<br><br></div>
    <div id="wynik" align="center">
        <script type="text/javascript">
            let a = prompt("Podaj pierwszą liczbę: ");
            let b = prompt("Podaj drugą liczbę: ");
            document.write("Liczby parzyste z przedziału (" + a + ", " + b + ")<br>");
            let i;
            var liczby="";
            if (a<b)
            {
                for (i=a; i<=b; i++)
                {
                    if (i%2==0)
                    {
                        liczby = liczby+i+" ";
                    }
                }
                document.write("Wynik: "+liczby);
            }
        </script>
    </div>
</body>
</html>
```

### Wykonaj zadanie:

Zmodyfikuj skrypt w powyższym przykładzie tak, aby poprawnie wyświetlał liczby, gdy zakres jest odwrócony (malejący) oraz gdy użytkownik poda te same liczby oraz gdy wprowadzone dane nie są liczbami. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_parzyste.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

## Temat: Pętla while w JavaScript.

### Zadanie:

Zapoznaj się z opisem pętli while w języku JavaScript publikowanym na witrynie internetowej serwisu w3schools: [https://www.w3schools.com/js/js\\_loop\\_while.asp](https://www.w3schools.com/js/js_loop_while.asp)

**Pętla while** zawiera instrukcje, które będą wykonane, gdy zostanie spełniony warunek pętli (wartość true).

Ogólna składnia pętli while:

```
while (warunek)           //nagłówek pętli
{
    instrukcje1;          //ciało pętli
}
```

Przykład zastosowania pętli while:

```
<script>
let i=1;
while (i <=10)           //nagłówek pętli
{
    document.write("Stan licznika pętli: " + i + "<br>");
    i++;
}
</script>
```

Inny przykład zastosowania pętli while:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
<title>Skrypty JavaScript</title>
<meta charset="utf-8">
<style>
    table, tr, td
    {
        border: 1px solid green;
    }
    table
    {
        text-align: center;
        width: 300px;
    }
</style>
</head>
<body>
<div id="instrukcja" align="center">Skrypt wyświetlający liczby losowe z przedziału
od 1 do 49.<br><br></div>
<div id="wynik" align="center">
<script type="text/javascript">
let i = 1;
let a;
document.write("<table><tr>");
while (i<=6)
{
    a = parseInt(Math.random()*49+1);
    document.write("<td width=50px>"+a+"</td>");
    i++;
}
document.write("</tr></table>");
</script>
</div>
</body>
</html>
```

Wykonaj zadanie:

Zmodyfikuj skrypt w powyższym przykładzie tak, aby wyświetlał 5 zestawów liczb losowych w kolejnych wierszach tabeli. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_losowe.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

**Pętla do while** zawiera instrukcje, które będą wykonane co najmniej raz, a następnie zostanie sprawdzony warunek pętli (wartość true decyduje o dalszym wykonaniu instrukcji). Ogólna składnia pętli do while:

```
do
```

```
{
  instrukcje1;
}
while (warunek);
```

Przykład zastosowania pętli do while:

```
<script>
let i=1;
do
  {
    //ciało pętli
    document.write("Stan licznika pętli: " + i + "<br>");
    i++;
  }
while (i <=10); //nagłówek pętli
</script>
```

Inny przykład zastosowania pętli do while:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
  <title>Skrypty JavaScript</title>
  <meta charset="utf-8">
  <style>
    table, tr, td
    {
      border: 1px solid green;
    }
    table
    {
      text-align: center;
      width: 400px;
    }
  </style>
</head>
<body>
  <div id="instrukcja" align="center">Skrypt wyświetlający liczby losowe z przedziału
od 1 do 49.<br><br></div>
  <div id="wynik" align="center">
    <script type="text/javascript">
      let i = 1;
      let a;
      document.write("<table>");
      do
      {
        document.write("<tr>");
        let j = 1;
        document.write("<td width=100px>Zestaw "+i+"</td>");
        do
        {
          a = parseInt(Math.random()*49+1);
          document.write("<td width=50px>"+a+"</td>");
          j++;
        }
        while (j<=6);
        document.write("</tr>");
        i++;
      }
      while (i<=5);
      document.write("</table>");
    </script>
  </div>
</body>
</html>
```

Wykonaj zadanie:

Napisz skrypt osadzony w pliku html w języku JavaScript wyświetlający tabliczkę mnożenia dla liczb od 1 do 10 w tabeli. W skrypcie zastosuj treści informujące o temacie zadania oraz tabelę z obramowaniem w kolorze niebieskim. Pracę zapisz w pliku pod nazwą `$nazwisko_$klasa_$gr_tabliczka.html` i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

## Temat: Funkcje w JavaScript.

### Zadanie:

Zapoznaj się z opisem funkcji w języku JavaScript publikowanym na witrynie internetowej serwisu w3schools: [https://www.w3schools.com/js/js\\_functions.asp](https://www.w3schools.com/js/js_functions.asp)

W języku JavaScript istnieje wiele funkcji wbudowanych, np.:

- `prompt()` – wyskakujące okno tekstowe,
- `alert()` – wyskakujące okno informacyjne,
- `confirm()` – wyskakujące okno decyzyjne,
- `parseInt()` – konwersja argumentu na liczbę,
- `parseFloat()` – konwersja argumentu na liczbę zmiennoprzecinkową,
- `Number()` – konwersja argumentu na liczbę,
- `isNaN()` – sprawdzenie, czy argument jest liczbą.

Dodatkowo możemy również definiować własne funkcje. Ogólna postać funkcji:

```
function nazwa_funkcji(arg1, ..., argn) //nagłówek
{
    instrukcje; //ciało
}
nazwa_funkcji(arg1, ..., argn) //wywołanie funkcji
```

### Przykład zastosowania funkcji:

```
<script>
    document.write("Skrypt obliczający sumę dwóch liczb.<br>");
    function suma(a,b)
    {
        document.write("Suma liczb "+a+" oraz "+b+" wynosi: "+(a+b));
    }
    suma(4,9);
</script>
```

### Inny przykład zastosowania funkcji:

```
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <title>Skrypty JavaScript</title>
</head>
<body>
    <div id="instrukcja">Skrypt obliczający sumę dwóch liczb podanych przez
    użytkownika.<br><br></div>
    <label for="liczba1">Podaj pierwszą liczbę:</label>
    <input type="number" id="liczba1"><br><br>
    <label for="liczba2">Podaj drugą liczbę:</label>
    <input type="number" id="liczba2"><br><br>
    <!--przycisk uruchamiający funkcję-->
    <button onclick="suma()">Oblicz sumę</button>
    <div id="wynik"></div>
<script>
    function suma()
    {
        var a = parseFloat(document.getElementById('liczba1').value);
        var b = parseFloat(document.getElementById('liczba2').value);
        document.getElementById('wynik').innerText = ("Suma liczb "+a+" oraz "+b+"
    wynosi: "+(a+b));
```

```
    }  
</script>  
</body>  
</html>
```

### Inny przykład zastosowania funkcji:

```
<!DOCTYPE HTML>  
<html lang="pl">  
<head>  
  <title>Skrypty JavaScript</title>  
  <meta charset="utf-8">  
  <script type="text/javascript">  
    function zegar()  
    {  
      let dzisiaj = new Date();  
      let dzien = dzisiaj.getDate();  
      if (dzien<10) { dzien = "0"+dzien};  
      let miesiac = dzisiaj.getMonth()+1;  
      let rok = dzisiaj.getFullYear();  
      let godzina = dzisiaj.getHours();  
      if (godzina<10) { godzina = "0"+godzina};  
      let minuta = dzisiaj.getMinutes();  
      if (minuta<10) { minuta = "0"+minuta};  
      let sekunda = dzisiaj.getSeconds();  
      if (sekunda<10) { sekunda = "0"+sekunda};  
      document.getElementById("czas").innerHTML  
        = dzien+"/"+miesiac+"/"+rok+" | "+godzina+": "+minuta+": "+sekunda;  
      setTimeout("zegar()",1000);  
    }  
  </script>  
</head>  
<body onload="zegar();">  
  <div id="instrukcja">Skrypt wyświetlający bieżący czas.<br><br></div>  
  <div id="czas"></div>  
</body>  
</html>
```

### Wykonaj zadanie:

Zmodyfikuj skrypt w powyższym przykładzie tak, aby zamiast numeru miesiąca wyświetlana była jego nazwa. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_zegar.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

### Inny przykład zastosowania funkcji:

```
<!DOCTYPE HTML>  
<html lang="pl">  
<head>  
  <title>Skrypty JavaScript</title>  
  <meta charset="utf-8">  
  <script type="text/javascript">  
    function sprawdz()  
    {  
      let liczba = document.getElementById("pole").value;  
      if (liczba>0) document.getElementById("wynik").innerHTML="Podana liczba jest  
      dodatnia.";  
      else if (liczba<0) document.getElementById("wynik").innerHTML="Podana liczba  
      jest ujemna.";  
      else if (liczba==0) document.getElementById("wynik").innerHTML="Podana liczba  
      jest równa zero.";  
      else document.getElementById("wynik").innerHTML="Podana wartość nie jest  
      liczbą.";  
    }  
  </script>  
</head>  
<body>  
  <div id="instrukcja">Skrypt sprawdzający znak podanej przez użytkownika  
  liczby.<br><br></div>
```



```
    <input type="text" id="pole"/>
    <input type="submit" value="Sprawdź" onclick="sprawdz()" />
    <div id="wynik"></div>
</body>
</html>
```

### Inny przykład zastosowania funkcji:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
  <title>Skrypty JavaScript</title>
  <meta charset="utf-8">
  <script type="text/javascript">
    function drukuj()
    {
      let a = document.getElementById("liczba1").value;
      let b = document.getElementById("liczba2").value;
      let zakres = "";
      for (i=a; i<=b; i++)
      {
        zakres = zakres+i+" ";
      }
      document.getElementById("wynik").innerHTML = zakres;
    }
  </script>
</head>
<body onload="slajdy()">
  <div id="instrukcja">Skrypt wyświetlający liczby z zakresu liczb podanych przez
  użytkownika.<br><br></div>
  <input type="text" id="liczba1"/>
  <input type="text" id="liczba2"/>
  <input type="submit" value="Pokaż" onclick="drukuj()" />
  <div id="wynik"></div>
</body>
</html>
```

### Inny przykład zastosowania funkcji:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
  <title>Skrypty JavaScript</title>
  <meta charset="utf-8">
  <script type="text/javascript">
    function drukuj()
    {
      let a = Number(document.getElementById("liczba1").value);
      let b = Number(document.getElementById("liczba2").value);
      let i;
      var liczby="";
      if (a<b)
      {
        for (i=a; i<=b; i++)
        {
          if (i%2==0)
          {
            liczby = liczby+i+" ";
          }
        }
        document.getElementById("wynik").innerHTML = liczby;
      }
      else if (a>b)
      {
        for (i=a; i>=b; i--)
        {
          if (i%2==0)
          {
            liczby = liczby+i+" ";
          }
        }
      }
    }
  </script>
</head>
<body>
  <div id="instrukcja">Skrypt wyświetlający liczby z zakresu liczb podanych przez
  użytkownika.<br><br></div>
  <input type="text" id="liczba1"/>
  <input type="text" id="liczba2"/>
  <input type="submit" value="Pokaż" onclick="drukuj()" />
  <div id="wynik"></div>
</body>
</html>
```

```
    }
  }
  document.getElementById("wynik").innerHTML = liczby;
}
else if (a==b)
{
  document.getElementById("wynik").innerHTML = "Liczby a i b są równe i nie ma przedziału";
}
else
{
  document.getElementById("wynik").innerHTML = "Podano niewłaściwe dane.";
}
}
</script>
</head>
<body>
  <div id="instrukcja">Skrypt wyświetlający liczby parzyste z przedziału liczb
  podanych przez użytkownika.<br><br></div>
  <input type="text" id="liczba1"/>
  <input type="text" id="liczba2"/>
  <input type="submit" value="Pokaż" onclick="drukuj()" />
  <div id="wynik" align="center"></div>
</body>
</html>
```

### Inny przykład zastosowania funkcji:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Skrypty JavaScript</title>
</head>
<body>
  <div align="center" width=300px>
    <h2>Kalkulator</h2>
    <h5>Skrypt wykonuje proste działania arytmetyczne.</h5>
    <label for="liczba1">Liczba 1:</label>
    <input type="number" id="liczba1"><br><br>
    <label for="liczba2">Liczba 2:</label>
    <input type="number" id="liczba2"><br><br>
    <label for="dzialanie">Wybierz operację:</label>
    <select id="dzialanie">
      <option value="add">Dodawanie</option>
      <option value="subtract">Odejmowanie</option>
      <option value="multiply">Mnożenie</option>
      <option value="divide">Dzielenie</option>
    </select><br><br>
    <!--przycisk uruchamiający funkcję-->
    <button onclick="oblicz()">Oblicz</button>
    <h3>Wynik: <span id="wynik"></span></h3>
  </div>
  <script>
    function oblicz()
    {
      var liczba1 = parseFloat(document.getElementById('liczba1').value);
      var liczba2 = parseFloat(document.getElementById('liczba2').value);
      var dzialanie = document.getElementById('dzialanie').value;
      let wynik;
      switch (dzialanie)
      {
        case 'add':
          wynik = liczba1+liczba2;
          break;
        case 'subtract':
          wynik = liczba1-liczba2;
          break;
        case 'multiply':
```

```
        wynik = liczba1*liczba2;
        break;
    case 'divide':
        if (liczba2 !== 0)
        {
            wynik = Math.round((liczba1/liczba2)*100)/100;
        } else
        {
            wynik = 'Nie można dzielić przez zero';
        }
        break;
    default:
        wynik = 'Nieznana operacja';
    }
    document.getElementById('wynik').innerText = wynik;
}
</script>
</body>
</html>
```

Wykonaj zadanie:

Zmodyfikuj skrypt w powyższym przykładzie tak, aby poprawić jego estetykę i działanie. Pracę zapisz w pliku pod nazwą **\$nazwisko\_\$klasa\_\$gr\_kalkulator.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

### Temat: Funkcje iteracyjne i rekurencyjne w JavaScript.

Funkcje iteracyjne wykonują instrukcje według chronologii zapisania w kodzie. Funkcja rekurencyjna (metoda rekurencyjna, rekursywna) to taka, która wywołuje samą siebie. W rekurencji funkcja rozkłada problem na mniejsze części, a następnie rozwiązuje każdą z nich przez wywołanie samej siebie. Samowywołanie się funkcji jest wywołaniem normalnym, gdzie wszystkie lokalne zmienne są tworzone oddzielnie w każdym wywołaniu, mechanizm przekazywania argumentów jest taki sam jak dla funkcji nierekursywnych.

Przykład zastosowania funkcji iteracyjnej do obliczania potęgi liczby 2:

```
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <title>Skrypty JavaScript</title>
</head>
<body>
    <div id="instrukcja">Skrypt obliczający potęgę liczby 2 podaną przez
    użytkownika.<br><br></div>
<script>
    function iteracyjna(x)
    {
        var wynik=1;
        if (x==0) return 1;
        else
        {
            for (var i=1;i<=x;i++)
            {
                wynik*=2;
            }
        }
        return wynik; //zakończenie działania funkcji, jeżeli wystąpią po tym kolejne
        instrukcje, to nie zostaną one wykonane
    }
    var n=parseInt(prompt("Podaj potęgę liczby 2: "));
    if (n>0)
    {
        document.write("Iteracyjnie 2 do potęgi "+n+" wynosi "+(iteracyjna(n))+"<br>");
    }
    else
    {
```

```
        document.write("Potęga nie może być liczbą ujemną.");
    }
</script>
</body>
</html>
```

Przykład zastosowania funkcji rekurencyjnej do obliczania potęgi liczby 2:

```
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <title>Skrypty JavaScript</title>
</head>
<body>
    <div id="instrukcja">Skrypt obliczający potęgę liczby 2 podaną przez
    użytkownika.<br><br></div>
    <label for="number">Podaj potęgę liczby 2:</label>
    <input type="number" id="number"><br><br>
    <button onclick="drukuj()">Oblicz potęgę rekurencyjnie</button>
    <div id="rekurencja"></div>
<script>
    function rekurencja(n)
    {
        if (n==0)
        {
            return 1;
        }
        else if (n>0)
        {
            var wynik=2*rekurencja(n-1);
            return wynik;
        }
        return 1/rekurencja(-n);
    }
    function drukuj()
    {
        var n = parseFloat(document.getElementById('number').value);
        document.getElementById('rekurencja').innerText = ("Rekurencyjnie 2 do potęgi
"+n+" wynosi "+rekurencja(n));
    }
</script>
</body>
</html>
```

## Temat: Obiekty w JavaScript.

Zadanie:

Zapoznaj się z opisem typu obiektowego publikowanym na witrynie internetowej serwisu w3schools:  
[https://www.w3schools.com/js/js\\_objects.asp](https://www.w3schools.com/js/js_objects.asp)

Obiekty to konstrukcje programistyczne, które posiadają właściwości zdefiniowane w zmiennych lub innych obiektach. Z obiektami powiązane są metody (funkcje) wykonujące operacje na właściwościach obiektu. Obiekt definiujemy przez funkcję zwaną konstruktorem oraz operatorem new. W celu wywołania właściwości danego obiektu stosujemy zapis: nazwa\_obiektu.nazwa\_właściwości, np. window.close().

W języku JavaScript istnieje wiele wbudowanych obiektów, np.:

- window – dostępne metody:
  - alert() – informacyjne okno dialogowe,
  - confirm() – okno decyzyjne,
  - prompt() – okno z możliwością wprowadzenia danych,
  - close() – zamknięcie okna z dokumentem,
  - open() – otwiera okno przeglądarki.
- document – dostępne metody i właściwości:

- write() – wypisanie podanego komunikatu w dokumencie,
  - clear() – wyczyszczenie zawartości okna,
  - close() – zamknięcie bieżącego okna,
  - getElementById(x) – odniesienie do elementu o podanym identyfikatorze,
  - getElementByName(x) – pobranie listy elementów o podanej nazwie,
  - getElementByTagName(tag) – pobranie listy elementów utworzonych za pomocą znacznika tag,
  - document.title = "Obiekty w JavaScript" – nowy tytuł dokumentu,
  - location – adres URL bieżącej strony,
  - document.bgColor = "gray" – szary kolor tła strony,
  - fgColor – kolor tekstu na stronie,
  - linkColor – kolor odnośnika,
  - vlinkColor – kolor użytego odnośnika,
  - alinkColor – kolor aktywnego odnośnika.
- location – dostępne metody:
- assign() – wczytuje dokument o podanym adresie URL,
  - reload() – ponowne wczytanie strony,
  - replace() – zastępuje bieżącą stronę wskazanym adresem URL.
- string – dostępne metody:
- tekst.length – zwraca liczbę znaków w łańcuchu,
  - tekst.big() – zwiększenie rozmiaru czcionki,
  - tekst.small() – zmniejszenie rozmiaru czcionki,
  - tekst.bold() – tekst pogrubiony,
  - tekst.italics() – tekst pochylony,
  - tekst.strike() – tekst przekreślony,
  - tekst.blink() – migający tekst,
  - tekst.fixed() – działanie znacznika <tt>,
  - tekst.sup() – indeks górny,
  - tekst.sub() – indeks dolny,
  - tekst.fontcolor("green") – kolor czcionki,
  - tekst.fontSize("20px") – ustawienie rozmiaru czcionki,
  - tekst.toUpperCase() – zamienia znaki na duże litery,
  - tekst.substring(5,10) – zwraca podany fragment łańcucha.
- Date – dostępne metody (new Date()):
- getYear() – rok,
  - getMonth() – numer miesiąca (0 – 11),
  - getDate() – dzień miesiąca (0 – 31),
  - getDay() – dzień tygodnia (0 – 6),
  - getHours() – godzina,
  - getMinutes() – minuta,
  - getSeconds() – sekunda,
  - setDate() – ustawia dzień miesiąca,
  - setHours(), setMinutes(), setSeconds(), setYear(),
  - toLocalString() – zwraca datę w postaci lokalnej.
- math – dostępne metody i właściwości:
- PI – stała wartość pi,
  - SQRT – stała wartość pierwiastek kwadratowy z 2,
  - SQRT1\_2 – stała wartość pierwiastek kwadratowy z 1/2,
  - abs(x) – wartość bezwzględna z x,
  - sqrt(x) – pierwiastek kwadratowy z x,
  - pow(x,y) – obliczenie x do potęgi y,
  - ceil(x) – najmniejsza liczba całkowita większa lub równa x,
  - floor(x) – największa liczba całkowita większa lub równa x,
  - max(x,y) – wyznaczenie większej liczby z x i y,
  - min(x,y) – wyznaczenie mniejszej liczby z x i y,
  - round(x) – zaokrąglenie liczby x do liczby całkowitej,

- `random()` – liczba pseudolosowa z przedziału  $[0,1)$ ,  $(\text{random()}*49+1)$ .
- `navigator` – dostępne metody:
  - `appName` – zwraca nazwę kodową przeglądarki,
  - `appNme` – zwraca nazwę przeglądarki,
  - `appVersion` – zwraca wersję przeglądarki,
  - `language` – zwraca język przeglądarki,
  - `userAgent` – zwraca nagłówek agenta przeglądarki,
- `history` – dostępne metody:
  - `length` – liczba adresów URL w historii strony,
  - `back()` – ładuje poprzedni adres URL z historii,
  - `forward()` – ładuje następny adres URL z historii,
  - `go()` – ładuje określony adres URL z historii.

Przykład zastosowania obiektów wbudowanych:

```
<script>
  console.log(document.location.href);
  document.bgColor = "black";
  document.fgColor = "silver";
  document.write("Dzisiejsza data to: "+(new Date()+"<br>"));
  document.write("Pierwiastek kwadratowy z liczby 5 wynosi:
"+(Math.round(Math.sqrt(5)*1000)/1000)+"<br>");
  document.write("Liczba 2 do potęgi 16 wynosi: "+Math.pow(2,16)+"<br>");
</script>
```

Dodatkowo możemy definiować własne obiekty. Przykłady tworzenia zmiennej typu `object`:

```
//definiowanie nowego obiektu car z trzema właściwościami
var car = {type:"Fiat", model:"500", color:"white"};

//definicja zmiennej typu Object o nazwie person
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

//zastosowanie zapisu wielowierszowego
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};

//utworzenie pustego obiektu i dodanie kolejnych właściwości
var person = {};
person.firstName = "John";
person.lastName = "Doe";
person.age = 50;
person.eyeColor = "blue";

//tworzenie obiektu z wykorzystaniem słowa kluczowego new
const person = new Object();
//dodanie właściwości i przypisanie im wartości
person.firstName = "John";
person.lastName = "Doe";
person.age = 50;
person.eyeColor = "blue";

//dostęp do właściwości obiektów w JavaScript
document.write(person.firstname + " is " + person.age + " years old.");
lub
document.write(person["firstname"] + " is " + person["age"] + " years old.");
lub
let x = "firstname";
let y = "age";
person[x] + " is " + person[y] + " years old.";

//dostęp do elementów obiektu
```

```
document.getElementById("wynik").innerHTML = "The last name is " + person.lastName;

//Metody obiektowe JavaScript
//Metody to akcje, które można wykonać na obiektach.
//Metody to definicje funkcji przechowywane jako wartości właściwości.
const person = {
  firstName: "John",
  lastName : "Doe",
  id      : 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};

//dostęp do metody obiektu (nazwa obiektu potem "." następnie nazwa właściwości)
document.write(person.fullName());

//dodanie metody do obiektu
person.name = function () {
  return this.firstName + " " + this.lastName;
};

//korzystanie z metod JavaScript
person.name = function () {
  return (this.firstName + " " + this.lastName).toUpperCase();
}

//zmiana wartości właściwości obiektu
const person = {
  firstName:"John",
  lastName:"Doe",
  age:50,
  eyeColor:"blue"
}
person.age = 21;

//dodanie nowej właściwości
person.nationality = "English";

//usunięcie właściwości
delete person.age;
lub
delete person["age"];

//definiowanie obiektów zagnieżdżonych (obektu w obiekcie)
myObj = {
  name:"John",
  age:30,
  myCars: {
    car1:"Ford",
    car2:"BMW",
    car3:"Fiat"
  }
}

//dostęp do wartości zagnieżdżonych
myObj.myCars.car2;
myObj.myCars["car2"];
myObj["myCars"]["car2"];
lub
let p1 = "myCars";
let p2 = "car2";
myObj[p1][p2];
```

#### Przykład definicji nowego obiektu:

```
<p id="wynik"></p>
<script>
  //Create an Object:
```

```
const car = {type:"Fiat", model:"500", color:"white"};
//Display Data from the Object:
document.getElementById("wynik").innerHTML = "The car type is " + car.type;
</script>
```

## Temat: Tablice w JavaScript.

Obiekt ARRAY służy do modyfikowania tablic, które przechowują wartości wielu zmiennych. Nową tablicę stworzymy za pomocą poleceń:

```
var nazwa_tablicy = new Array();
var nazwa_tablicy = [];
typeof nazwa_tablicy;           //object (typ obiektowy)
```

### Zadanie:

Zapoznaj się z opisem typu tablicowego publikowanym na witrynie internetowej serwisu w3schools:  
[https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp)

### Przykłady tworzenia zmiennej typu Array:

```
//tworzenie tablicy z wykorzystaniem słowa kluczowego new
//taka tablica musi zawierać minimum dwa elementy
var cars = new Array("Saab", "Volvo", "BMW");

//definicja zmiennej typu tablicowego, może zawierać jeden element
var cars = ["Saab", "Volvo", "BMW"];

//zastosowanie zapisu wielowierszowego
var cars = [
  "Saab",
  "Volvo",
  "BMW"
];

var cars = [];
cars[0]= "Saab";
cars[1]= "Volvo";
cars[2]= "BMW";

//dostęp do elementów tablicy
var cars = ["Saab", "Volvo", "BMW"];
let car = cars[0];
console.log(cars[1]);

//zmiana elementu w tablicy
var cars = ["Saab", "Volvo", "BMW"];
cars[0] = "Opel";

//konwertowanie tablicy na łańcuch znaków
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.toString();

//dostęp do całej tablicy
var cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars;

//tablica jest szczególnym typem obiektu
var person = ["John", "Doe", 46];
document.getElementById("demo").innerHTML = person[0];
var person = {firstName:"John", lastName:"Doe", age:46};
document.getElementById("demo").innerHTML = person.firstName;

//długość tablicy odpowiada ilości elementów
var fruits = ["Banana", "Orange", "Apple", "Mango"];
let size = fruits.length;
document.getElementById("demo").innerHTML = size;

//dostęp do ostatniego elementu tablicy
```



```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits[fruits.length-1];

//wylistowanie elementów tablicy
var fruits = ["Banana", "Orange", "Apple", "Mango"];
let fLen = fruits.length;
let text = "<ul>";
for (let i = 0; i < fLen; i++)
{
  text += "<li>" + fruits[i] + "</li>";
}
text += "</ul>";
document.getElementById("demo").innerHTML = text;

//zastosowanie pętli for of do wylistowania elementów tablicy
var fruits = ["Banana", "Orange", "Apple", "Mango"];
let text = "<ul>";
for (let fruit of fruits)
{
  text += "<li>" + fruit + "</li>";
}
text += "</ul>";
document.getElementById("wynik").innerHTML = text;

//wylistowanie elementów tablicy za pomocą funkcji Array.forEach()
var fruits = ["Banana", "Orange", "Apple", "Mango"];
let text = "<ul>";
fruits.forEach(myFunction);
text += "</ul>";
document.getElementById("demo").innerHTML = text;
function myFunction(value) {
  text += "<li>" + value + "</li>";
}

//dodanie kolejnego elementu do tablicy
var fruits = ["Banana", "Orange", "Apple"];
fruits.push("Lemon"); // Adds a new element (Lemon) to fruits
lub
var fruits = ["Banana", "Orange", "Apple"];
fruits[fruits.length] = "Lemon"; // Adds "Lemon" to fruits

//w JavaScript tablice zawsze używają indeksów numerowanych
var person = [];
person[0] = "John";
person[1] = "Doe";
person[4] = 46; //elementy 2 i 3 tablicy będą traktowane jako "undefined"
person.length; // Will return 5
person[0]; // Will return "John"

//definiowanie obiektów zagnieżdżonych (tablica w obiekcie)
myObj = {
  name:"John",
  age:30,
  myCars: ["Saab", "Volvo", "BMW"];
}
//dostęp do elementów tablicy zapisanej w obiekcie
console.log(myObj.myCars[0]);
```

### Przykład zastosowania obiektu Array:

```
<script>
  var dniTygodnia = new Array('Niedziela', 'Poniedziałek', 'Wtorek', 'Środa',
'Czwartek');
  document.write(dniTygodnia[3]);
  document.write('<br>'+dniTygodnia.length);
  dniTygodnia[6]='Sobota';
  document.write('<br>'+dniTygodnia.length);
  document.write('<br>'+dniTygodnia[6]);
```

```
document.write('<br>'+dniTygodnia[5]);
dniTygodnia[5]='Piątek';
document.write('<br>Kolejne dni tygodnia z tablicy:<br>');
for (var i=0;i<=6;i++)
{
    document.write(' '+dniTygodnia[i]);
}
</script>
```

### Inny przykład zastosowania tablicy:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
    <title>Skrypty JavaScript</title>
    <meta charset="utf-8">
    <style>
        table, tr, td
        {
            border: 2px solid green;
        }
        td
        {
            width: 50px;
        }
        table
        {
            width: 300px;
            height: 30px;
            text-align: center;
            border-collapse: collapse;
        }
    </style>
</head>
<body>
    <div id="instrukcja" align="center">Skrypt sumujący 5 wartości tablicy podanej
    przez użytkownika.<br><br></div>
    <div id="wynik" align="center">
    <script>
        var liczby=new Array(5);    //deklaracja pięcioelementowej tablicy, "undefined"
        var suma = 0;
        for (var i=0;i<=4;i++)
        {
            liczby[i]=parseInt(prompt("Podaj liczbę: "));
            suma = suma+liczby[i];
        }
        document.write('Podane następujące liczby do tablicy:');
        document.write('<table><tr>');
        for (j=0;j<=4;j++)
        {
            document.write('<td>'+liczby[j]+'</td>');
        }
        document.write('</tr></table>');
        document.write('<br>Suma liczb w tablicy wynosi: '+suma);
    </script>
    </div>
</body>
</html>
```

### Inny przykład zastosowania tablicy zagnieżdżonej (tablica w tablicy):

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
    <title>Skrypty JavaScript</title>
    <meta charset="utf-8">
</head>
<body>
```

```
<div id="instrukcja"><h2>Skrypt prezentujący zagnieżdżone obiekty i tablice
JavaScript.</h2></div>
<div id="wynik"></div>
<script>
  let x = "";
  var myObj = {
    name: "John",
    age: 30,
    cars:
      [
        {name:"Ford", models:["Fiesta", "Focus", "Mustang"]},
        {name:"BMW", models:["320", "X3", "X5"]},
        {name:"Fiat", models:["500", "Panda"]}
      ]
  }
  for (let i in myObj.cars) {
    x += "<h2>" + myObj.cars[i].name + "</h2>";
    for (let j in myObj.cars[i].models) {
      x += myObj.cars[i].models[j] + "<br>";
    }
  }
  document.getElementById("wynik").innerHTML = x;
</script>
</body>
</html>
```

## Temat: Obsługa zdarzeń w JavaScript.

Zdarzenia na stronie można podzielić na:

- zdarzenia myszy,
- zdarzenia klawiatury,
- zdarzenia dokumentu,
- zdarzenia formularza.

Wybrane zdarzenia:

- onBlur – usunięcie aktywności pola,
- onChange – zmiana wartości,
- onclick – kliknięcie znacznika,
- ondblclick – podwójne kliknięcie znacznika,
- onerror – błąd podczas ładowania,
- onfocus – uaktywnienie elementu,
- onkeydown – wciśnięcie i przytrzymanie klawisza,
- onkeypress – krótkotrwałe wciśnięcie klawisza,
- onkeyup – zwolnienie klawisza,
- onload – ładowanie strony,
- onmousedown – wciśnięcie i przytrzymanie klawisza myszy,
- onmouseout – opuszczenie danego obszaru przez kursor myszy,
- onmouseover – najechanie kursorem myszy na dany obszar,
- onmouseup – puszczenie klawisza myszy,
- onmove – poruszanie oknem,
- onreset – wyczyszczenie formularza,
- onresize – zmiana rozmiaru okna,
- onselect – wybranie danego elementu,
- onsubmit – zatwierdzenie formularza,
- onunload – zamknięcie strony.

Przykład wykorzystania zdarzeń:

```
<!DOCTYPE HTML>
<html lang="pl">
```

```
<head>
  <title>Skrypty JavaScript</title>
  <meta charset="utf-8">
  <script type="text/javascript">
    function koloruj(x)
    {
      if (x==1)
      {
        document.fgColor="blue";
      }
      if (x==2)
      {
        document.fgColor="red";
      }
    }
  </script>
</head>
<body onload="podaj_imie()">
  <div id="instrukcja" align="center">Skrypt obsługujący zdarzenie najechania mysza
na obszar.<br><br></div>
  <div id="wynik" align="center">
    <h1 onmouseover="koloruj(1)" onmouseout="koloruj(2)">Przykładowy tekst.</h1>
  </div>
</body>
</html>
```

Wykonaj zadanie:

Utwórz skrypt wyświetlający różne napisy po kliknięciu zdjęć w tabeli jednowymiarowej. Pracę zapisz w pliku pod nazwą **\$nazwisko\_ \$klasa\_ \$gr\_ onclick.html** i prześlij do nauczyciela w postaci załącznika na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).

### Temat: Formularze w JavaScript.

Formularze na stronach www służą do zbierania informacji od użytkownika odwiedzającego stronę. W języku HTML formularze tworzymy wewnątrz znacznika `<form>`. Każdy element formularza stanowi odrębny obiekt z własnymi właściwościami. Odwołanie do formularza uzyskujemy w następujący sposób:

```
document.forms[0] //pierwszy formularz
document.nazwa_formularza //formularz o zdefiniowanej nazwie (name="nazwa")
document.forms[1].elements[0] //pierwszy element drugiego formularza
document.nazwa_formularza.nazwa_elementu //konkretny element konkretnego formularza
document.forms[1].elements[0].name
document.forms[1].elements[0].value
```

Zadanie:

Zapoznaj się z opisem typu tablicowego publikowanym na witrynie internetowej serwisu w3schools: [https://www.w3schools.com/js/js\\_validation.asp](https://www.w3schools.com/js/js_validation.asp)

Atrybuty znacznika `<form>` (<https://webkod.pl/kurs-html/tagi/formularz/element-form>):

- name – nazwa przypisana do formularza,
- action="https://greszata.pl/wyslij.php" –
- method – get (domyślnie), post – metody przekazania zawartości formularza,
- onsubmit –
- accept-charset="utf-8" – kodowanie znaków, domyślnie ustawienie dokumentu,
- autocomplete="on" – autouzupełnianie wpisywanych wartości,
- enctype="application/urlencoded" – sposób kodowania danych,
- novalidate – wartości formularza nie są wymagane,
- target="\_self" (\_blank) – miejsce wczytania adresu docelowego dla formularza (bieżące / nowe okno).

Atrybuty znacznika `<input>` (<https://webkod.pl/kurs-html/tagi/formularz/element-input>):

- name – nazwa elementu,
- type – text, submit, file, number, botton, checkbox, color, date, e-mail, image, password, radio, range, reset...

- required – wartość wymagana, gdy będzie pusta zostanie wyświetlony błąd przeglądarki,
- value – wartość wyświetlana,
- max – maksymalna wartość (liczba, data...),
- min – minimalna wartość(liczba, data...),
- maxlength – maksymalna liczba znaków,
- minlength – minimalna liczba znaków,
- size="5" – liczba widocznych znaków w polu,
- heigth="30" – wysokość elementu,
- pattern="[0-9]{4}" – wzorzec elementu wejściowego (cztery znaki od 0 do 9),
- disabled – wyłączenie elementu wejściowego,
- alt – alternatywny podpis elementu,
- list="lista" – rozwijalna lista elementów (<datalist id="lista"><option>1</option></datalist>),
- readonly – element tylko do odczytu (data, telefon, url...),
- src – plik graficzny reprezentujący element,
- title – tekst wyświetlany w przypadku błędnego podania danych.

Atrybuty znacznika <button> (<https://webkod.pl/kurs-html/tagi/formularz/element-button>):

- name – nazwa elementu,
- type – submit,

Przykład wykorzystania formularza:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
  <title>Skrypty JavaScript</title>
  <meta charset="utf-8">
  <script>
    function powitanie(form1)
    {
      if (form1.imie.value=="")
      {
        document.write('Nie podałeś swojego imienia.');
      }
      else
      {
        document.write('Witaj '+form1.imie.value);
      }
    }
  </script>
</head>
<body>
  <div id="instrukcja" align="center">Skrypt pobierający imie z
formularza.<br><br></div>
  <div id="wynik" align="center"></div>
  <form method="post" name="form1">
    Podaj swoje imie:
    <input type="text" name="imie">
    <input type="button" name="przycisk" value="Kliknij"
onClick="powitanie(this.form)">
  </form>
</body>
</html>
```

Inny przykład wykorzystania formularza:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
  <title>Skrypty JavaScript</title>
  <meta charset="utf-8">
  <script>
    function pobierz()
    {
```

```
f=document.forms[0]; //pierwszy element tablicy formularzy
napis=" \n";
braki=false;
if (f.imie.value=="")
{
    napis+="imie\n";
    braki=true;
}
if (f.nazwisko.value=="")
{
    napis+="nazwisko\n";
    braki=true;
}
if (f.data.value=="")
{
    napis+="data";
    braki=true;
}
if (braki==false)
{
    alert("Wprowadziłeś wszystkie dane.");
}
else
{
    alert("Brakuje następujących danych:\n"+napis);
}
</script>
</head>
<body>
<div id="instrukcja" align="center">Skrypt pobierający dane użytkownika z
formularza.<br><br></div>
<form method="post" name="form">
<table>
<tr><td>Podaj swoje imię:</td>
<td><input type="text" name="imie"></td></tr>
<tr><td>Podaj swoje nazwisko:</td>
<td><input type="text" name="nazwisko"></td></tr>
<tr><td>Podaj rok urodzenia:</td>
<td><input type="text" name="data" size="4"></td></tr>
<tr><td colspan="2" align="right"><input type="button" name="przycisk"
value="Kliknij" onClick="pobierz()"></td></tr>
</table>
</form>
</body>
</html>
```

### Inny przykład wykorzystania formularza:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
<title>Skrypty JavaScript</title>
<meta charset="utf-8">
<style>
#wynik
{
    background-color: #00ff00;
    width: 100px;
    text-align=: center;
    color: blue;
    border: 2px solid red;
    border-radius: 5px;
}
</style>
<script>
function oblicz()
{
    var a=Number(document.getElementById("liczba1").value);
```

```
        var b=Number(document.getElementById("liczba2").value);
        var c=document.getElementById("dzialanie").value;
        if (c=="suma") document.getElementById("wynik").innerHTML=a+b;
        if (c=="roznica") document.getElementById("wynik").innerHTML=a-b;
        if (c=="iloczyn") document.getElementById("wynik").innerHTML=a*b;
        if (c=="iloraz") document.getElementById("wynik").innerHTML=a/b;
    }
</script>
</head>
<body>
  <div id="instrukcja" align="center">Skrypt pobierający dane z formularza oraz
  wykonujący działania arytmetyczne.<br><br></div>
  <form method="post" name="form">
    <table align="center">
      <tr><td><input type="text" id="liczba1"></td>
      <td><select id="dzialanie">
        <option value="suma">+</option>
        <option value="roznica">-</option>
        <option value="iloczyn">*</option>
        <option value="iloraz">/</option>
      </select></td>
      <td><input type="text" id="liczba2"></td>
      <td>=</td>
      <td id="wynik"></td></tr>
      <tr><td colspan="5">
        <input type="button" value="Oblicz" onClick="oblicz()">
      </td></tr>
    </table>
  </form>
</body>
</html>
```

#### Polecane strony:

<https://podstawyjs.pl/spis-tresci/>  
<https://www.w3schools.com/js/default.asp>  
<https://codepen.io/pen/?editors=0010>  
<https://jsfiddle.net/>  
<https://www.jshint.com/>  
<https://www.youtube.com/watch?v=OcwON22ctYc>

#### UWAGA!

W razie problemów kieruj pytania do nauczyciela na adres [greszata@zs9elektronik.pl](mailto:greszata@zs9elektronik.pl).